Chapter 6
Configure Logical Interface Properties

For a physical interface device to function, you must configure at least one logical interface on that device. For each logical interface, you must specify the protocol family that the interface supports. You can also configure other logical interface properties. These vary by PIC and encapsulation type, but include the IP address of the interface, whether or not the interface supports multicast traffic, DLCIs, VCIs and VPIs, and traffic shaping.

To configure logical interface properties, you include the unit statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    accounting-profile name;
    bandwidth rate;
    disable;
    dlci dlci-identifier;
    description text;
    drop-timeout milliseconds;
    encapsulation type;
    family {
        protocol-family-statements;
    }
    fragment-threshold bytes;
    inverse-arp;
    minimum-links number;
    mrru bytes;
    multicast-dlci dlci-identifier;
    multicast-vci vpi-identifier.vci-identifier;
    multipoint;
    no-traps;
    oam-liveness {
        up-count cells;
        down-count cells;
    }
    oam-period (disable | seconds);
    passive-monitor-mode;
    point-to-point;
    shaping {
        (cbr rate | vbr peak rate sustained rate burst length);
        queue-length number;
    }
```

```
        short-sequence;
        tunnel {
            source source-address;
            destination destination-address;
                routing-instance {
                    destination routing-instance-name;
                }
                ttl number;
        }
        vci vpi-identifier.vci-identifier;
        vlan-id number;
    }
```

This chapter describes the logical interface properties that you can configure:

Specify the Logical Interface Number on page 103

Add a Logical Unit Description to the Configuration on page 103

Configure a Point-to-Point Connection on page 103

Configure a Multipoint Connection on page 104

Configure Accounting on page 104

Configure the Interface Bandwidth on page 105

Disable SNMP Notifications on Logical Interfaces on page 106

Configure Interface Encapsulation on page 106

Disable a Logical Interface on page 108

Configure a Point-to-Point Frame Relay Connection on page 108

Configure a Drop Timeout Period on page 108

Configure Multilink Encapsulation on page 109

Configure a Fragmentation Threshold on page 110

Configure Inverse ATM ARP on page 110

Configure Inverse Frame Relay ARP on page 111

Configure Multilink Minimum Links on page 111

Configure MRRU on page 112

Configure a Multicast-Capable Frame Relay Connection on page 112

Configure a Multicast-Capable ATM Connection on page 113

Configure a Point-to-Multipoint ATM Connection on page 113

Configure the ATM OAM F5 Loopback Cell Threshold on page 114

Define the ATM OAM F5 Loopback Cell Period on page 114

Enable Passive Monitoring on page 115

Define the ATM Traffic-Shaping Profile on page 116

Configure Sequence Format on page 119

Configure Tunnel Interfaces on page 119

Configure 802.1Q VLAN IDs on page 122

Configure a Point-to-Point ATM Connection on page 122

Configure the Maximum Number of VCs on a VP on page 123

Table 9 lists statements that you can use to configure logical interfaces.

Table 9: Statements for Logical Interface Properties

Statement	Interface Types	Usage Guidelines
accounting-profile *name*	All.	"Configure Accounting" on page 104.
bandwidth *rate*	All interface types except multilink and aggregated.	"Configure the Interface Bandwidth" on page 105.
destination (*destination-address* | *routing-instance-name*)	Encryption, GRE tunnel, and IP tunnel interfaces.	"Configure a Unicast Tunnel" on page 120, "Configure a VPN Loopback Tunnel for Route Table Lookup" on page 120, or "Configure the Tunnel Address for an Encryption Interface" on page 121.
description	All.	"Add a Logical Unit Description to the Configuration" on page 103.
disable	All.	"Disable a Logical Interface" on page 108.
dlci *dlci-identifier*	Interfaces with Frame Relay encapsulation.	"Configure a Point-to-Point Frame Relay Connection" on page 108.
drop-timeout *milliseconds*	Multilink interfaces.	"Configure a Drop Timeout Period" on page 108.
encapsulation *type*	All interface types except aggregated SONET/SDH and loopback.	"Configure the Encapsulation on a Logical Interface" on page 106.
family	All.	"Configure the Protocol Family" on page 127.
fragment-threshold *bytes*	Multilink interfaces.	"Configure a Fragmentation Threshold" on page 110.
inverse-arp	Interfaces with ATM or Frame Relay encapsulation.	"Configure Inverse ATM ARP" on page 110 or "Configure Inverse Frame Relay ARP" on page 111.
minimum-links *number*	Multilink interfaces.	"Configure Multilink Minimum Links" on page 111.
mrru *bytes*	Multilink interfaces.	"Configure MRRU" on page 112.
multicast-dlci *dlci-identifier*	Point-to-multipoint Frame Relay interfaces.	"Configure a Multicast-Capable Frame Relay Connection" on page 112.

Statement	Interface Types	Usage Guidelines
multicast-vci *vpi-identifier.vci-identifier*	Point-to-multipoint ATM interfaces.	"Configure the ATM OAM F5 Loopback Cell Threshold" on page 114.
multipoint	All.	"Configure a Multipoint Connection" on page 104.
no-traps	All.	"Disable SNMP Notifications on Logical Interfaces" on page 106.
oam-liveness	Interfaces with ATM encapsulation.	"Configure the ATM OAM F5 Loopback Cell Threshold" on page 114.
oam-period (disable | *seconds*)	Interfaces with ATM encapsulation.	"Define the ATM OAM F5 Loopback Cell Period" on page 114.
passive-monitor-mode	SONET interfaces.	"Enable Passive Monitoring" on page 115.
point-to-point	All.	"Configure a Point-to-Point Connection" on page 103.
queue-length *number*	Interfaces with ATM encapsulation.	"Define the ATM Traffic-Shaping Profile" on page 116.
routing-instance	GRE tunnel and IP tunnel interfaces.	"Configure a Unicast Tunnel" on page 120 or "Configure a VPN Loopback Tunnel for Route Table Lookup" on page 120.
shaping	Interfaces with ATM encapsulation.	"Define the ATM Traffic-Shaping Profile" on page 116.
short-sequence	Multilink interfaces.	"Configure Sequence Format" on page 119.
source *address*	Encryption, GRE tunnel, and IP tunnel interfaces.	"Configure a Unicast Tunnel" on page 120, "Configure a VPN Loopback Tunnel for Route Table Lookup" on page 120, or "Configure the Tunnel Address for an Encryption Interface" on page 121.
ttl *number*	GRE tunnel and IP tunnel interfaces.	"Configure a Unicast Tunnel" on page 120.
tunnel	Encryption, GRE tunnel, and IP tunnel interfaces.	"Configure a Unicast Tunnel" on page 120, "Configure a VPN Loopback Tunnel for Route Table Lookup" on page 120, or "Configure the Tunnel Address for an Encryption Interface" on page 121.
vci *vpi-identifier.vci-identifier*	ATM point-to-point interfaces.	"Configure a Point-to-Point ATM Connection" on page 122.
vlan-id *number*	Fast Ethernet and Gigabit Ethernet interfaces.	"Configure 802.1Q VLAN IDs" on page 122.

Specify the Logical Interface Number

Each logical interface must have a logical unit number. The logical unit number corresponds to the logical unit part of the interface name. For more information, see “Configure the Interface Name” on page 38.

PPP and Cisco HDLC encapsulations support only a single logical interface, whose logical unit number must be 0. Frame Relay and ATM encapsulations support multiple logical interfaces, so you can configure one or more logical unit numbers.

You specify the logical unit number in the unit statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces]
interface-name {
    unit 0 {
        ...
    }
}
interface-name {
    unit logical-unit-number {
        ...
    }
}
```

The logical unit number can range from 0 through 16384.

Add a Logical Unit Description to the Configuration

You can include a text description of each logical unit in the configuration file. Any descriptive text you include is displayed in the output of the show interfaces commands. It has no impact on the interface’s configuration. To add a text description, include the description statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
description text;
```

The description can be a single line of text. If the text contains spaces, enclose it in quotation marks.

For information about describing physical interfaces, see “Add an Interface Description to the Configuration” on page 40.

Configure a Point-to-Point Connection

By default, all interfaces are assumed to be point-to-point connections. You must ensure that the MTU sizes on both sides of the connection are the same.

For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection by including the point-to-point statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
point-to-point;
```

Configure a Multipoint Connection

By default, all interfaces are assumed to be point-to-point connections. To configure an interface to be a multipoint connection, include the multipoint statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multipoint;
```

Configure Accounting

Juniper Networks routers can collect various kinds of data about traffic passing through the router. You can set up one or more *accounting pr ofiles* that specify some common characteristics of this data, including the following:

- The fields used in the accounting records
- The number of files that the router retains before discarding, and the number of bytes per file
- The period that the system uses to record the data

You configure the profiles using statements in the accounting-options hierarchy; for more information, see the *JUNOS Internet Softw are Configur ation Guide: Netw ork Management* . You must assign a unique name for each accounting profile; this name cross-references the information specified in the accounting-options hierarchy with interfaces or firewall configuration statements.

Configure the Logical Interface Profile

There are two types of accounting profiles: interface profiles and filter profiles. They have different configuration statements in the accounting-profiles hierarchy, and are implemented separately in either the interfaces or firewall hierarchy. If you reference the same profile from both a firewall filter and an interface statement within the same configuration, it causes an error.

The following is a sample accounting-options profile for an interface; for more information, see the *JUNOS Internet Software Configuration Guide: Network Management*:

```
[edit]
accounting-options {
    file if_stats {
        size 4m files 10 transfer-interval 15;
        archive-sites {
            "ftp://login:password@host/path";
        }
    }
    interface-profile if_profile {
        interval 15;
        file if_stats {
            fields {
                input-bytes;
                output-bytes;
                input-packets;
                output-packets;
                input-errors;
                output-errors;
            }
        }
    }
}
```

Generally, you configure interface profiles at the physical interface level; however, you can also reference them to a particular logical circuit within the routing network. To enable accounting on a specific circuit, include the accounting-profile statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
accounting-profile name1;
unit 0 {
    accounting-profile name2;
}
```

To reference profiles by physical interface, see "Configure Physical Interface Profiles" on page 55. For information about configuring a firewall filter accounting profile, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Configure the Interface Bandwidth

By default, the JUNOS software uses the physical interface's speed for the MIB-II object, ifSpeed. You can configure the logical unit to populate the ifSpeed variable by configuring a bandwidth value for the logical interface. The bandwidth statement sets an informational-only parameter; you cannot adjust the actual bandwidth of an interface with this statement.

To configure the bandwidth value for a logical interface, include the bandwidth statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
bandwidth rate;
```

rate is the peak rate, in bps or cps. You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000). You can also specify a value in cells per second by entering a decimal number followed by the abbreviation c; values expressed in cells per second are converted to bits per second using the formula 1 cps = 384 bps. The range is not limited. The bandwidth statement is valid for all logical interfaces, except multilink and aggregated interfaces.

Disable SNMP Notifications on Logical Interfaces

By default, SNMP notifications are sent when the state of an interface or a connection changes. To disable these notifications on the logical interface, include the no-traps statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
no-traps;
```

Configure Interface Encapsulation

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, you must configure an encapsulation to use for packets transmitted on the interface. For more information about physical interface encapsulation, see "Configure the Encapsulation on a Physical Interface" on page 44.

You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

Configure the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for some encapsulation types, such as Frame Relay, ATM, and Ethernet VLAN encapsulations, you also can configure the encapsulation type that is used inside the Frame Relay, ATM, or VLAN circuit itself. To do this, include the encapsulation statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation (atm-ccc-cell-relay | atm-ccc-vc-mux | atm-tcc-vc-mux | atm-cisco-nlpid | atm-nlpid |
atm-snap | atm-tcc-snap | atm-vc-mux | ether-over-atm-llc | frame-relay-ccc | frame-relay-tcc |
multilink-ppp | multilink-framerelay | vlan-ccc);
```

Some of the ATM encapsulations are defined in RFC 1483, *Multiprotocol Encapsulation o ver ATM Adaptation Layer 5*.

With the atm-nlpid, atm-cisco-nlpid, and atm-vc-mux encapsulations, you can configure the family inet only. With the circuit cross-connect (CCC) circuit encapsulations, you cannot configure a family on the logical interface. A logical interface cannot have frame-relay-ccc encapsulation unless the physical device also has frame-relay-ccc encapsulation. A logical interface cannot have frame-relay-tcc encapsulation unless the physical device also has frame-relay-tcc encapsulation. In addition, you must assign this logical interface a DLCI in the range 512 through 1022 and configure it as point-to-point.

For interfaces that carry IPv4 traffic, you can configure ether-over-atm-llc encapsulation. When you use ether-over-atm-llc encapsulation, you cannot configure multipoint interfaces.

A logical interface cannot have vlan-ccc encapsulation unless the physical device also has vlan-ccc encapsulation. For encapsulation type vlan-ccc, you must assign this logical interface a VLAN ID in the range 512 through 1023; if the VLAN ID is 511 or lower, it is subject to the normal destination filter lookups in addition to source address filtering.

For information about multilink encapsulations, see “Configure Multilink Encapsulation” on page 109.

You can create an ATM cell relay circuit by configuring an entire ATM physical device or an individual virtual circuit (VC). When you configure an entire device, only cell relay encapsulation is allowed on the logical interfaces; you control the number and location of VCs using the atm-options statement. Allowed VCs on both ingress and egress ATM interfaces should be the same. For most interfaces, you can define a maximum of 4090 VCs per interface. The highest numbered VC value you can configure is 4089. For ATM OC-3 interfaces, you can define a maximum of 8186 VCs per interface. For ATM OC-12 interfaces, you can define a maximum of 16,378 VCs per interface. Promiscuous mode removes these limits. For more information, see “Configure ATM Cell-Relay Promiscuous Mode” on page 175.

If you are dedicating the entire device to a cell relay circuit, enter the allow_any_vci statement under unit 0 as shown in the following example. Once you enter this statement, you cannot configure other logical interfaces in the same physical interface.

```
[edit interfaces at-1/2/0]
encapsulation atm-ccc-cell-relay;
atm-options {
    vpi 0 maximum-vcs 256;
}
unit 0 {
    point-to-point;
    encapsulation atm-ccc-cell-relay;
    allow_any_vci;
}
```

Alternatively, to configure an individual VC on a specific logical interface, include a statement similar to the following example:

```
[edit interfaces at-1/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
    vpi 0 maximum-vcs 256;
}
unit 120 {
    encapsulation atm-ccc-cell-relay;
    vci 0.120;
}
```

Note: When you use ATM CCC cell relay encapsulation, you must configure both the physical and logical encapsulation with atm-ccc-cell-relay. You cannot mix different logical encapsulation types on an interface that you have configured with ATM CCC cell relay physical encapsulation.

Disable a Logical Interface

You can unconfigure a logical interface, effectively disabling that interface, without removing the logical interface configuration statements from the configuration. To do this, include the disable statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
disable;
```

Configure a Point-to-Point Frame Relay Connection

To configure a point-to-point Frame Relay connection, include the dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
dlci dlci-identifier;
```

dlci-identifier is the DLCI identifier, which is a number from 1 through 1022. Numbers 1 to 15 are reserved. A point-to-point interface can have one DLCI.

Note

Channelized DS-3 and Channelized STM-1 to E1 interfaces can support a maximum of 64 DLCIs per channel, for a total of 1792 DLCIs per DS-3 interface or 4032 DLCIs per STM-1 interface. For more information, see "Configure Channelized DS-3 to DS-1 Interfaces" on page 197 or "Configure Channelized STM-1 Interfaces" on page 206.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connection must be the same.

For more information, see "Configure Frame Relay" on page 251.

Configure a Drop Timeout Period

By default, the drop timeout parameter is disabled. You can configure a drop timeout value to provide an enhanced recovery mechanism in case individual links in the multilink bundle drop one or more packets. Make sure any value you set is larger than the expected differential delay across the links, although drop timeout is not a differential delay tolerance setting, and does not limit the overall latency.

To configure a drop timeout value, include the drop-timeout statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
drop-timeout milliseconds;
```

milliseconds is the duration of the drop timer; its range is 1 through 127 ms. Values less than 5 ms are not recommended; a value of 0 disables the timer.

Note

For multilink interfaces, if a packet or fragment encounters an error condition and it is destined for a disabled bundle or link, it does not contribute to the dropped packet and frame counts in the per-bundle statistics. The packet is counted under the global error statistics and is not included in the global output bytes and output packet counts. This unusual accounting happens only if the error conditions are generated inside the multilink interface, not if the packet encounters errors on the wire or elsewhere in the system.

For more information, see "Configure Multilink Interfaces" on page 261.

Configure Multilink Encapsulation

By default, the encapsulation on multilink interfaces is MLPPP. MLPPP and MLFR are the encapsulation types used to transmit packets within the multilink circuit; for general information on this parameter, see "Configure the Encapsulation on a Logical Interface" on page 106.

You can leave the encapsulation statement set to its default value of multilink-ppp or change the value to multilink-framerelay.

To configure multilink encapsulation, include the encapsulation statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
encapsulation (multilink-ppp | multilink-framerelay);
```

For more information, see "Configure Multilink Interfaces" on page 261.

Configure a Fragmentation Threshold

By default, the fragmentation threshold parameter is disabled. You can configure a fragmentation threshold to set a maximum size for packet payloads transmitted across the individual links within the multilink circuit. The software splits any incoming packet that exceeds the fragmentation threshold into smaller units suitable for the circuit size; it reassembles the fragments at the other end, so this feature does not affect the output traffic stream. The threshold value refers to the payload only; it does not include the MLPPP header.

> **Note**
>
> To ensure proper load-balancing:
>
> For MLFR interfaces, do not include the fragmentation-threshold statement in the configuration.
>
> For MLPPP interfaces, do not include both the fragmentation-threshold statement and the short-sequence statement in the configuration.
>
> For both MLFR and MLPPP interfaces, if the MTU of links in a bundle is less than the bundle MTU plus encapsulation overhead, then fragmentation is automatically enabled. You should avoid this situation for MLFR interfaces and for MLPPP interfaces with short-sequencing enabled.

To configure a fragmentation threshold value, include the fragment-threshold statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
fragment-threshold bytes;
```

bytes is the maximum fragment size, beyond which the software automatically subdivides packet payloads; its range is 128 through 16,320 bytes. Any value you set must be a multiple of 64 bytes (*N*x64). The default value of 0 results in no fragmentation.

For more information, see “Configure Multilink Interfaces” on page 261.

Configure Inverse ATM ARP

You can configure ATM interfaces to support inverse ATM ARP, as described in RFC 2225. When inverse ATM ARP is enabled, the router responds to received inverse ATM ARP requests by providing IP address information to the requesting ATM device.

The router does not initiate inverse ATM ARP requests.

By default, inverse ATM ARP is disabled. To configure a VC to respond to inverse ATM ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* multipoint-destination *destination*] hierarchy level:

```
inverse-arp;
```

You must configure ATM LLC-SNAP encapsulation on the logical interface to support inverse ARP. The other ATM encapsulation types are disallowed. For more information, see "Configure Interface Encapsulation" on page 106.

For more information, see "Configure ATM Interfaces" on page 173.

Configure Inverse Frame Relay ARP

Frame Relay interfaces support inverse Frame Relay ARP, as described in RFC 2390. When inverse Frame Relay ARP is enabled, the router responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting router on the other end of the Frame PVC (permanent virtual circuit).

The router does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a router to respond to inverse Frame Relay ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* multipoint-destination *destination*] hierarchy level:

```
inverse-arp;
```

You must configure Frame Relay encapsulation on the logical interface to support inverse ARP. For more information, see "Configure Interface Encapsulation" on page 106.

For more information, see "Configure Frame Relay" on page 251.

Configure Multilink Minimum Links

You can set the minimum number of links that must be up for the multilink bundle as a whole to be labeled up. To set the minimum number, include the minimum-links statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
minimum-links number;
```

By default, minimum-links has a value of 1. *number* can be a value from 1 through 8.

For more information, see "Configure Multilink Interfaces" on page 261.

Configure MRRU

The maximum received reconstructed unit (MRRU) is similar to a maximum transmission unit (MTU), but applies to multilink bundles; it is the maximum packet size that the multilink interface can process. By default, the MRRU is set to 1500 bytes; you can configure a different MRRU value if the peer equipment allows. The MRRU includes the original payload plus the 2-byte PPP header, but not the additional MLPPP or MLFR header applied while the individual multilink packets are traversing separate links in the bundle.

To configure a different MRRU value, include the mrru statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
mrru bytes;
```

bytes is the MRRU size; its range is 1500 through 4500 bytes.

> **Note**
>
> If you set the MRRU on a bundle to a value larger than the MTU of the individual links within it, you must enable a fragmentation threshold for that bundle. Set the threshold to a value no larger than the smallest MTU of any link included in the bundle.
>
> Determine the appropriate MTU size for the bundle by ensuring the MTU size does not exceed the sum of the encapsulation overhead and the MTU sizes for the links in the bundle.

For more information, see “Configure Multilink Interfaces” on page 261.

Configure a Multicast-Capable Frame Relay Connection

By default, Frame Relay connections assume unicast traffic. If your Frame Relay switch performs multicast replication, you can configure the connection to support multicast traffic by including the multicast-dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-dlci dlci-identifier;
```

dlci-identifier is the DLCI identifier, which is a number from 1 through 1022 that defines the Frame Relay DLCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint Frame Relay connections.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For information about disabling keepalives, see “Configure Keepalives” on page 50.

For more information, see “Configure Frame Relay” on page 251.

Configure a Multicast-Capable ATM Connection

By default, ATM connections assume unicast traffic. If your ATM switch performs multicast replication, you can configure the connection to support multicast traffic by including the multicast-vci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-vci vpi-identifier.vci-identifier;
```

vci-identifier and *vpi-identifier* are the VCI and VPI identifiers, which define the ATM VCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint ATM connections.

For more information, see “Configure ATM Interfaces” on page 173.

Configure a Point-to-Multipoint ATM Connection

To configure a point-to-multipoint (NBMA) ATM connection, include the multipoint-destination statement within the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *inet* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet]
address address {
    multipoint-destination destination-address vci vpi-identifier.vci-identifier;
}
```

address is the interface’s address. The address must include the destination prefix (for example, /24).

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *vci-identifier* and *vpi-identifier* are the VCI and optional VPI identifiers for the connection.

When you configure point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

For more information, see “Configure ATM Interfaces” on page 173.

Configure the ATM OAM F5 Loopback Cell Threshold

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell threshold on VCs, which is the minimum number of consecutive OAM F5 loopback cells received before declaring that a VC is up or lost before declaring that a VC is down.

By default, when five consecutive OAM F5 loopback cells are received, the VC is considered to be up, and when five consecutive cells are lost, the VC is considered to be down. To modify these values on a point-to-point interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    vci vpi-identifier.vci-identifier;
    oam-liveness {
        up-count cells;
        down-count cells;
    }
}
```

To modify the OAM F5 loopback cell count threshold on a virtual circuit that is part of a point-to-multipoint interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
    multipoint-destination destination-address {
        vci vpi-identifier.vci-identifier;
        oam-liveness {
            up-count cells;
            down-count cells;
        }
    }
}
```

The cell count can be a value from 1 through 255 cells.

For more information, see “Configure ATM Interfaces” on page 173.

Define the ATM OAM F5 Loopback Cell Period

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell period on virtual circuits, which is the interval at which OAM F5 loopback cells are transmitted.

By default, no OAM F5 loopback cells are sent. To send OAM F5 loopback cells on a point-to-point interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    vci vpi-identifier.vci-identifier;
    oam-period (disable | seconds);
}
```

To send OAM F5 loopback cells on a virtual circuit that is part of a point-to-multipoint interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
   multipoint-destination destination-address {
      vci vpi-identifier.vci-identifier;
      oam-period (disable | seconds);
   }
}
```

The period can range from 1 through 900 seconds. You can also enter oam-period disable, which disables the OAM loopback cell transmit feature.

OAM VC-AIS (alarm indication signal) and VC-RDI (remote defect indication) cells are used for identifying and reporting VC defects end-to-end. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream VCs affected by the failure. Upon receiving an AIS cell on a VC, the router marks the logical interface down and sends an RDI cell on the same VC to let the remote end know the error status. When an RDI cell is received on a VC, the router sets the logical interface status to down. When no AIS or RDI cells are received for 3 seconds, the router sets the logical interface status to up. You do not need to configure anything to enable defect indication.

For more information, see “Configure ATM Interfaces” on page 173.

Enable Passive Monitoring

The passive monitoring PIC is one of a group of multiservice PICs, specifically designed to enable IP services. If you have Passive Monitoring PICs and SONET/SDH PICs installed in an M160 or M40e router, you can monitor IPv4 traffic from another router. On SONET interfaces, you enable packet flow monitoring by including the passive-monitor-mode statement at the [edit interfaces so-*fpc*/*pic*/*port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces so-fpc/pic/port unit logical-unit-number]
passive-monitor-mode;
```

If you include the passive-monitor-mode statement in the configuration, the SONET interface does not send keepalives or alarms, and does not participate actively on the network.

On passive monitoring interfaces, you enable packet flow monitoring by including the family statement at the [edit interfaces mo-*fpc*/*pic*/*port* unit *logical-unit-number*] hierarchy level, specifying the inet option:

```
[edit interfaces mo-fpc/pic/port unit logical-unit-number]
family inet;
```

For the passive monitoring interface, you can configure multiservice physical interface properties. For more information, see “Configure Multiservice Physical Interface Properties” on page 81.

Define the ATM Traffic-Shaping Profile

When you are using an ATM encapsulation, you can configure a traffic-shaping profile that defines the following:

- Bandwidth utilization, which consists of either a constant rate, or a peak cell rate with sustained cell rate and burst tolerance
- Maximum queue length

These values are used in the ATM generic cell-rate algorithm, which is a leaky bucket algorithm that defines the short-term burst rate for ATM cells, the maximum number of cells that can be included in a burst, and the long-term sustained ATM cell traffic rate. Each individual VC has its own independent shaping parameters.

By default, the bandwidth utilization is unlimited; that is, unspecified bit rate (UBR) is used. Also, by default, buffer usage by VCs is unregulated. To define limits to bandwidth utilization on a point-to-point interface or to limit buffer use, include the shaping statement. For point-to-point interfaces, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
shaping {
    (cbr rate | vbr peak rate sustained rate burst length);
    queue-length number;
}
```

For virtual circuits that are part of a point-to-multipoint interface, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
shaping {
    (cbr rate | vbr peak rate sustained rate burst length);
    queue-length number;
}
```

When defining the ATM traffic-shaping profile, you can do the following:

- Configure CBR on page 116
- Configure VBR on page 117
- Specify Shaping Values on page 117

Configure CBR

For traffic that does not need to burst periodically to a higher rate, you can configure a constant bit rate (CBR) by including the cbr statement at the [edit interfaces *interface-name* unit *logical-unit-number* shaping] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* shaping] hierarchy level:

```
cbr rate;
```

For more information, see “Specify Shaping Values” on page 117.

Configure VBR

To define variable bandwidth rate (VBR) utilization, include the vbr statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
vbr peak rate sustained rate burst length;
```

You can define the following VBR traffic-shaping properties:

Peak rate—Top rate at which traffic can burst.

Sustained rate—Normal traffic rate averaged over time.

Burst length—Maximum number of cells that a burst of traffic can contain. It can be a value from 1 through 255 cells.

For more information, see "Specify Shaping Values" on page 117.

Specify Shaping Values

You can specify the rates in bits per second (bps) or cells per second (cps). For OC-3c interfaces, the highest rate is 135,600,000 bps (353,125 cps), which corresponds to 100 percent of the available line rate. For OC-12c interfaces, the highest rate is 271,263,396 bps (706,415.09 cps), which corresponds to 50 percent of the available line rate. Table 10 lists some of the other rates you can specify. If you specify a rate that is not listed, it is rounded to the nearest rate.

The exact number of values differs between OC-12c and OC-3c interfaces. OC-12c interfaces have about four times as many value increments as OC-3c interfaces. For OC-12c rates between 1/2 and 1/128 of the line rate, there are 128 steps between each 1/*n* value. For rates smaller than 1/128, there are (16,384 minus 128) or 16,256 values. This calculation is valid because fractional shaping is ignored at rates below 1/128. This results in about 32,384 distinct rates for OC-12c. For OC-3c, the starting point is full line rate, the fraction/integer breakpoint is about 1/32, and there is a maximum of 4096 scheduler slots, producing about 8032 distinct values.

In general, the actual packet rate on the interface is calculated with the following formula:

actual-rate = (128 * *line-rate*) / (trunc ((128 * *line-rate*) / *desired-rate*))

line-rate is the maximum available rate on the interface (in bits per second) after factoring out the overhead for SONET and ATM (per-cell) overheads. For OC-3c interfaces, the line rate is calculated as follows:

line-rate = 155,520,000 bps x (26/27) x (48/53) = 135,600,000 bps

For OC-12c interfaces, the line rate is calculated as follows:

line-rate = 622,080,000 bps x (26/27) x (48/53) = 542,526,792.45 bps

desired-rate is the rate you enter in the vbr statement, in bits per second.

The trunc operator indicates that all digits to the right of the decimal point should be dropped.

For OC-3c interfaces, the maximum available rate is 100 percent of *line-rate*, or 135,600,000 bps. For OC-12c interfaces, the maximum available rate is 50 percent of *line-rate*, or 271,263,396 bps.

The following example shows the calculations for determining the actual rate when the desired rate is 80 percent of the maximum rate:

OC-3c:
135,600,000 bps * 0.8 = 108,480,000 bps
actual-rate = (128 * 135,600,000.1) / (trunc ((128 * 135,600,000.1) / 108,480,000))
actual-rate = 17,356,800,013 / (trunc (17,356,800,013 / 108,480,000))
actual-rate = 17,356,800,013 / 160
actual-rate = 108,480,000 bps

OC-12c:
271,263,396 bps * 0.8 = 217,010,716.8 bps
actual-rate = (128 * 542,526,792.45) / (trunc ((128 * 542,526,792.45)/217,010,716.8))
actual-rate = 69,443,429,434 / (trunc (69,443,429,434 / 217,010,716.8))
actual-rate = 69,443,429,434 / 320
actual-rate = 217,010,717 bps

Table 10: Traffic-Shaping Rates

Interface Type	Line Rate (bps)	Line Rate (cps)	Percentage of Total Line Rate
OC-3			
	135,600,000	353,125	100.00
	134,542,320	350,370.66	99.22
	133,511,760	347,686.88	98.46
	132,494,760	345,038.44	97.71
	131,491,320	342,425.31	96.97
	130,501,440	339,847.5	96.24
	129,525,120	337,305	95.52
	128,562,360	334,797.81	94.81
	127,626,720	332,361.25	94.12
	126,691,080	329,924.69	93.43
OC-12			
	271,263,396	706,415.09	50.00
	270,207,897	703,666.40	49.81
	269,160,579	700,939.01	49.61
	268,121,349	698,232.68	49.42
	267,090,113	695,547.17	49.23
	266,066,779	692,882.24	49.04
	265,051,257	690,237.65	48.85
	264,043,458	687,613.17	48.67
	263,043,293	685,008.58	48.48
	262,050,677	682,423.64	48.30

Buffers are shared among all VCs, and by default, there is no limit to the buffer size for a VC. If a VC is particularly slow, it might use all the buffer resources. To limit the queue size of a particular VC, include the queue-length statement when configuring the VC at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
queue-length number;
```

The length can range from 1 through 16,383 packets. The default is 16,383 packets.

For more information, see “Configure ATM Interfaces” on page 173.

Configure Sequence Format

For MLPPP, the sequence header format is set to 24 bits by default. You can configure an alternative value of 12 bits, but 24 bits is considered the more robust value for most networks.

To configure a 12-bit sequence header, include the short-sequence statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
short-sequence;
```

For MLFR, the sequence header format is set to 12 bits by default. This is the only valid option.

For more information, see “Configure Multilink Interfaces” on page 261.

Configure Tunnel Interfaces

With a tunnel PIC installed in your platform, you can do the following:

Configure a Multicast Tunnel

For interfaces that carry IPv4 or IPv6 traffic, you can configure multicast tunnels. To configure a multicast tunnel, include the multicasts-only statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] or [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
multicasts-only;
```

For more information, see “Configure Tunnel Interfaces” on page 311.

Configure a Unicast Tunnel

To configure a unicast tunnel, you configure the gr interface (to use GRE encapsulation) or the ip interface (to use IP-IP encapsulation) and include the tunnel statement:

```
[edit interfaces]
gr-fpc/pic/port or ip-fpc/pic/port {
    unit logical-unit-number {
        tunnel {
            source address;
            destination address;
            routing-instance {
                destination routing-instance-name;
            }
            ttl number;
        }
        family family {
            address address {
                destination address;
            }
        }
    }
}
```

For more information, see "Configure Tunnel Interfaces" on page 311.

Configure a VPN Loopback Tunnel for Route Table Lookup

To configure tunnel interfaces to facilitate route table lookups for VPNs, you specify a tunnel's endpoint IP addresses and associate them with a routing instance that belongs to a particular routing table. This enables the software to search in the appropriate routing table for the route prefix, because the same prefix can appear in multiple routing tables. To configure the destination VPN, include the routing-instance statement at the [edit interfaces gr-*fpc*/*pic*/*port* unit *logical-unit-number* tunnel] hierarchy level:

```
[edit interfaces]
gr-fpc/pic/port {
    unit logical-unit-number {
        tunnel {
            source address;
            destination address;
            routing-instance {
                destination routing-instance-name;
            }
        }
    }
}
```

Note

You can configure a VPN loopback tunnel for either VRF table lookup or route table lookup, not both. For more information, see "Configure a VPN Loopback Tunnel for VRF Table Lookup" on page 121.

For more information, see "Configure Tunnel Interfaces" on page 311.

Configure a VPN Loopback Tunnel for VRF Table Lookup

To configure a tunnel interface to facilitate VPN routing and forwarding (VRF) table lookup based on MPLS labels, you specify a VPN loopback tunnel interface name and associate it with a routing instance that belongs to a particular routing table.

To specify a VPN loopback tunnel interface name, you configure the vt interface and include the family inet and family mpls statements:

```
[edit interfaces]
vt-fpc/pic/port {
    unit 0 {
        family inet;
        family mpls;
    }
    unit 1 {
        family inet;
    }
}
```

To associate the VPN loopback tunnel with a routing instance, include the VPN loopback tunnel interface name at the [edit routing-instances] hierarchy level:

```
[edit routing-instances] {
interface vt-fpc/pic/port;
```

> **Note**
> You can configure a VPN loopback tunnel for either VRF table lookup or route table lookup, not both. For more information, see "Configure a VPN Loopback Tunnel for Route Table Lookup" on page 120.

For the vt interface, none of the statements in the tunnel configuration block are valid.

For more information about VRF tables, see the see the *JUNOS Internet Softw are Configur ation Guide: VPNs* . For more information about tunnels, see "Configure Tunnel Interfaces" on page 311.

Configure the Tunnel Address for an Encryption Interface

Secure traffic travels through tunnel interfaces between remote hosts. You configure each IPSec tunnel as a logical interface on the ES PIC. As you do with other tunnel interfaces, include the tunnel statement at the [edit interfaces es-*fpc/pic/port* unit *logical-unit-number*] hierarchy level to specify the source and destination addresses:

```
[edit interfaces]
es-fpc/pic/port {
    unit logical-unit-number {
        tunnel {
            source address;
            destination address;
        }
    }
}
```

For more information, see "Configure Encryption Interfaces" on page 223.

Configure 802.1Q VLAN IDs

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, the software supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or broadcast domain.

Gigabit Ethernet interfaces can be partitioned; you can assign up to 4095 different logical interfaces, one for each VLAN, but you are limited to a maximum of 1024 VLANs on any single Gigabit Ethernet port. Fast Ethernet interfaces can also be partitioned, with a maximum of 1024 logical interfaces for the four-port FE PIC and 16 logical interfaces for the M40e and M160 FE-48 PIC. Table 11 lists VLAN ID range by interface type.

Table 11: VLAN ID Range by Interface Type

Interface Type	VLAN ID Range
4-port, 8-port, and 12-port Fast Ethernet	0 through 1023
48-port Fast Ethernet	0 through 4094
Gigabit Ethernet	0 through 4094
Management and internal Ethernet interfaces	0 through 1023

To bind a VLAN ID to a logical interface, include the vlan-id statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
vlan-id number;
```

For more information, see “Configure Ethernet Interfaces” on page 229.

Configure a Point-to-Point ATM Connection

To configure a VCI and a VPI on a point-to-point ATM interface, include the vci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
vci vpi-identifier.vci-identifier;
```

For each VCI, you configure the VCI and VPI identifiers. The default VPI identifier is 0. The VCI identifier cannot exceed the highest numbered VC configured for the interface with the vpi statement, as described in “Configure ATM Physical Interface Properties” on page 60.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same.

For more information, see “Configure ATM Interfaces” on page 173.

Configure the Maximum Number of VCs on a VP

When you are configuring ATM VCs, you can configure the maximum number of virtual circuits (VCs) allowed on a virtual path (VP). You configure this value so that sufficient memory on the ATM PIC can be allocated for each VC.

To configure the highest numbered VCs on a VP, include the atm-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
atm-options {
    vpi vpi-identifier maximum-vcs maximum-vcs;
}
```

The VP identifier can be a value from 0 through 255. For most interfaces, you can define a maximum of 4090 VCs per interface. The highest numbered VC value you can configure is 4089. For ATM OC-3 interfaces, you can define a maximum of 8186 VCs per interface. For ATM OC-12 interfaces, you can define a maximum of 16,378 VCs per interface. Promiscuous mode removes these limits. For more information, see “Configure ATM Interfaces” on page 173.

For more information, see “Configure ATM Interfaces” on page 173.